

2/1/97 - 1/31/98

User Interfaces for Cooperative Remote Design

Grant #
N00014-96-1-G004

Larry Hodges, William Ribarsky

GVU Center, Georgia Institute of Technology,
Atlanta GA 30332-0280

19981103081

We are now putting together and are demonstrating a prototype design environment. Based on Performer, OpenGL, and Multigen, it is flexible enough for intergration with simulations and interfaces supplied by Lockheed-Martin or DARPA. It can work in networked environments involving multiple workbenches, workbenches and displays, HMD systems and workbenches,or HMD systems and displays.

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

DTIC QUALITY INSPECTED 4



Home of the 1996 Olympic Village
**Georgia Institute
of Technology**

Office of Contract Administration
Contracting Support Division
Atlanta, Georgia 30332-0420 U.S.A.
PHONE 404-894-6944
FAX 404-894-5285

October 27, 1998

In reply refer to: **C-50-659**

Office of Naval Research
Program Officer, Dr. Lawrence Rosenblum
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217-5660

Subject: **Annual Technical Report**
Project Director: **Larry Hodges**
Telephone No.: **(404)894-8787**
Contract No.: **N00014-96-1-G004**
Prime No.: **N/A**
"USER INTERFACES FOR COOPERATIVE REMOTE DESIGN"
Period Covered: **970201 through 980131**

The subject report is forwarded in conformance with the contract/grant specifications.

Should you have any questions or comments regarding this report(s), please contact the Project Director or the undersigned at 404-894-4764.

/TW

Sincerely,

Wanda W. Simon
Wanda W. Simon
Reports Coordinator

Distribution:
Addressee: **3 copies**
1 copy to: Director, Naval Research Laboratory
2 copies to: Defense Technical Information Center
Cc:ONR-RR

COOPERATIVE REMOTE DESIGN
Annual Report
January 31, 1998

Our objective in this research has been to allow teams that are physically separated to do detailed design work on large-scale, 3D projects. We seek to develop new methods to facilitate cooperative remote design utilizing both high-bandwidth networking capability and virtual reality with appropriate graphical interfaces to support the collaborative effort. The specific task is to enable multiple users to stand at each of two (or more) Virtual Workbenches at remote locations and interact effectively for design generation. We have made significant progress towards this goal.

To build and test our design environment we needed a large scale and complex design model. We chose the Navy Arsenal Ship. This is a multiple deck model and our biggest initial problem was to organize and segment this model so that graphical detail could be handled efficiently and so that individual 3D objects could be moved independently. We separated the model hierarchically along decks and then further along task areas (e.g., galley, mess area, engine room, communications rooms, sleeping quarters, etc.). Initially, we implemented the model using the Simple Virtual Environment (SVE), which provides extensive software support for the creation of virtual applications including handling tracking, interactive tools, virtual menus, and display of 3D objects within the virtual environment. After this initial work we switched from SVE to the SGI Performer environment. The main reasons for this is that Performer is highly optimized for running 3D interactive simulations on Reality Engines, Infinite Reality Engines, and other high-end SGI hardware, and it has recently been made more flexible with new capabilities. Its optimizations including advanced handling of graphical details and textures to sustain interactivity and use of parallelism in multiprocessor architectures. Other capabilities include the handling of over 30 model formats and integration with Multigen modeling capacities.

After this initial work we have:

- Developed off-axis stereoscopic display and basic tracking mechanisms within the Performer environment.
- Developed an object data organization for the design environment. This involved labeling by base element, function, deck, size, modular grouping, and other attributes. With this structure we can locate and place objects, group them, evaluate their fit and use, and provide input to automatically update CAD databases.
- We have generated nearly 1,000 3D objects and classified them using the data organization format. These are items for the living quarters, galley, and mess areas, among others. We or others can now generate additional objects in Multigen, 3D studio, Wavefront, or other formats accepted by Performer or Multigen.
- Used the data organization, to develop models for a number of objects in the cargo and engine rooms, living quarters, mess hall, galley, passageways, and other areas on all decks of the ship. These have now been converted into the Multigen format.
- Worked on several interaction paradigms, many of which are implemented in the design environment. These include two-handed manipulation to grab, orient, and stretch or compress objects. Also, we have an interface for pinch gloves, whereby separate gestures involving one finger touching any other can be accurately recognized, and a serial interface for a tool handle that has multiple buttons and can be programmed for different purposes.
- Developed a networking structure that supports the collaborative virtual environment by allowing all objects in the environment to be "shared". By this we mean that each user in the VE will be able to see all objects (models) in the environment and any changes made to those objects. If one user moves a chair, the chair will move for all other

users, keeping each user's view of that chair (position, orientation, etc.) consistent with everyone else's. Users will also see graphical representations (avatars) of the other users in the environment. These avatars are themselves "shared" objects.

We have now put together and are demonstrating a prototype design environment. Based on Performer, OpenGL, and Multigen, it is flexible enough for integration with simulations and interfaces supplied by Lockheed-Martin or DARPA. It can work in networked environments involving multiple workbenches, workbenches and displays, HMD systems and workbenches, or HMD systems and displays. It allows:

- Participants to see representations (avatars) of all users interacting in the environment. The avatars will vary depending on the type of environment. For example, in the networked workbench environment, one might just see representations of collaborators' hands. In an environment involving a workbench and HMD, the person in the HMD would be immersed and might see only her collaborator's hand while the person at the workbench would see an avatar. The avatar would have the position and orientation of the immersed person in the HMD.
- Changes made to the model to be communicated to each participant allowing them to see an up-to-date representation of objects in the environment.
- Participants to communicate with each other through a digital voice channel. While this is not absolutely necessary, we believe that it greatly enhances the collaboration.
- Users to perform simple manipulations (translation, rotation, simple grouped actions) with the individual objects comprising the ship model.

FURTHER WORK

Now that we have our initial system in place, we will coordinate with the Naval Research Lab's simulation-based design efforts. This will include working with Division, which has been chosen to interface its dVISE virtual prototyping software with HLA. We will direct efforts towards integrating dVISE into our collaborative environment.

Georgia Tech has recently won a DURIP grant and has used this funding plus significant matching funds to obtain its own immersive workbench setup. This facility is compatible with the NRL system. We have now begun a joint study involving two workbenches and a long-distance, networked environment. We will further investigate setting up a link between DARPA and NRL for demonstration and further studies of collaboration. This will permit us to study an environment with more than two sites participating and with one or more collaborators at each site.

We will further study issues of collaboration. Working these out will be especially important when we have multiple, networked users. There must, for example, be support for "group memory" since the design projects will be spatially complex with multiple steps and multiple sessions. The collaborators must have support to recall what they did both in individual sessions and across sessions. In addition we must work out protocols for collaboration to reduce latency of interaction and induce efficient sharing of tools. For example, we will institute various roles for collaborators, such as "lecturer and audience", "apprentice and master", and "peers", among others. These roles range from total control of tools and simulation objects on one end to a free-for-all environment (peers) where one collaborator can take over control of a tool without permission. In addition to addressing control and style of work in different ways, these protocols have different impacts on latency since the imposing and removing of locks on tools (and the broadcasting of this information) takes time and can interrupt the flow of interaction.

In line with our study of the multi-user, collaborative environment and issues of data transfer and latency, we will investigate a robust networked structure. This will be based on work we are doing now, in conjunction with the Georgia Tech Telecommunications/

Networking Research Group and NRL. (Our present networked interface is based on this work.) The networked structure is based on the concepts of dynamically configurable protocols and adaptable applications. It will dynamically adjust to current network conditions, sacrificing speed for reliability (or vice-versa) based on the type of data. For example, for data from an electromagnetic tracker, it's important to transmit the data as fast as possible, but it's no big loss if an occasional transmission doesn't make it. Therefore the reliability of the transmission can be sacrificed in order to reduce the latency. However, suppose the actual structure of a model is changed? It's important that all users be notified of this change, making the reliability of the transmission more important than the latency. We will be able to configure the networking in our application to distinguish between these types of transmissions.

Finally we will continue our work in enlarging and managing the set of objects in the ship design. This will include further methods for grouping, re-grouping (including making new groups), and ungrouping of functional elements in the design space. For the more complicated objects and groups we will introduce multiple levels of detail, which can be handled in the Performer environment. We will also work on integrated detail management strategies involving cost/benefit analyses to maintain interactivity. (These are necessary when handling views containing many groups of objects.) We must also introduce dynamic objects, which allow changes to object structure or function in a simulation-based environment. This will necessitate efficient organization, passing, and incorporation of updates to object attributes (beyond position and orientation) between collaborators.

Interaction in Semi-Immersive Large Display Environments

Rogier van de Pol^{1,2}, William Ribarsky¹, Larry Hodges¹, and Frits Post²

¹GVU Center, Georgia Tech, ²Delft Technical University

Abstract

This paper evaluates interaction methods within the general framework of navigation, selection, and manipulation. It considers large display environments and, in particular, the virtual workbench, comparing this system with HMD and CAVE systems. The paper addresses three issues: (a) identifying the characteristics that set the workbench apart from other virtual environments; (b) determining types, organization, and examples of interaction techniques; (c) evaluating how these techniques perform on the workbench to determine which perform best. The evaluations are based on an extensive set of user observations. Also discussed are some problems that stereoscopic display coupled with interaction bring out.

I. Introduction

The availability of economical, bright, high resolution large displays, especially projected displays, has significantly increased their use. In particular this is true for tracked immersive or semi-immersive systems. CAVE-like immersive systems [3] with two, three, four or more displays are used in a variety of engineering, design, and scientific applications. In the last few years, the semi-immersive virtual workbench [5,8,12] with its horizontal or slanted display has come into use for engineering and scientific applications, often involving collaboration. In the future we can expect to see an even wider use of large displays, often with (frequently wireless) tracking. They will be used in "smart rooms" perhaps including the future personal office.

In spite of the interest in immersive and semi-immersive large display environments, there has been relatively little work that provides an overall classification and evaluation of interaction techniques for these environments. This lack is compounded because these environments tend to use stereoscopic display more frequently than, say, head mounted display (HMD) systems. Stereoscopic display coupled with interaction provides its own special problems, some of which are heightened in semi-immersive environments. For these reasons we address interaction in semi-immersive environments in this paper.

We provide a general classification of interaction techniques while addressing implementation in particular from the standpoint of semi-immersive systems such as the virtual workbench. For the workbench our goals are

- To find what the characteristics are for the workbench that set it apart from other environments, like HMD and CAVE systems
- To determine the types and organization of interaction techniques
- To see how these techniques perform in the workbench and to determine which might be best for that environment.

Some of the conclusions on uses of interaction, however, can be extended to other environments.

II. Related Work

Although the CAVE has been around longer than the virtual workbench and there have been several papers on the design of the environment [3] and on applications, there has been relatively little work looking in detail at the types and behavior of interaction. Recently there have been tools, such as CALVIN [9], developed for collaboration in systems such as the CAVE. These tools have been used on architectural design and other applications. But here again there is little evaluation of interaction methods. In the present paper several of the methods evaluated are applicable to the CAVE and other large display systems, not just to the workbench.

Recently, Cutler et. al. [2] focused on two-handed interaction in a workbench environment. Techniques were investigated to manipulate objects using both hands with Pinch Gloves as input devices. The described techniques adhered to Guiard's principles of two-handed tasks [6], where the non-dominant hand provides a reference for the dominant hand. The dominant hand performs the actual task. In this paper we discuss both one- and two-handed interaction, gaze-directed interaction, and input devices in addition to the Pinch Gloves. In addition we compare and contrast the workbench to other immersive and semi-immersive devices. The workbench is also a natural environment for collaboration; a recent study [1] looked at ways to allow two (or more) people to work together effectively on the workbench. This involved two people being head-tracked simultaneously by multiplexing all four eye views. We discuss a few alternative ideas that would not involve the need for simultaneous head-tracking.

A classification and description of interaction tasks in virtual environments is given by Mine [10], distinguishing navigation, selection, and manipulation as three general tasks. Bowman and Hodges discuss the need for constraints in virtual environments [2]. They consider both virtual constraints (software) and physical

constraints (e.g., a tabletop), giving examples of how to constrain interaction tasks to improve efficiency and accuracy. We use here a classification scheme similar to that of Mine in discussing interactions in large display environments and also discuss the role of constraints. The importance of exploiting the proprioceptive sense is discussed in Mine et. al. [11]. Here it is argued that because the user is often unable to see his hands, proprioception can help improve interaction. By bringing an object close to his body and by using intuitive hand gestures, the user has a better understanding of the position and orientation of the object. We think that proprioception is important in large display environments, even though the user is able to see his body, and we discuss how it affects the interactive methods we describe.

III. Types of Interaction Techniques

It is useful to provide a classification of techniques that can be used in a semi-immersive, large display environment. In this section we will do this and give a description of each.

Three universal tasks that can be found in almost any virtual reality application are [2]:

- Navigation
- Selection
- Manipulation

We have implemented some navigation techniques, but our main focus has been on selection and manipulation in the workbench environment. Furthermore, we have focused on direct interaction techniques, where the user directly interacts with objects using natural gestures (as opposed to using sliders or menus).

Navigation

In this paper we define navigation as the process of changing the position and orientation of the viewpoint in the environment. The most obvious navigation technique in a workbench environment is to let the user physically walk around the workbench to get different views of the scene. To accomplish this the user's head is tracked. Two other navigation techniques we implemented are *panning* and *scaling*. It can be argued that these are not really navigation techniques, since the user's viewpoint doesn't change. However, since these techniques affect the position of the entire scene relative to the viewpoint, we include them here.

Panning and Scaling. Panning is important if the virtual scene is larger than the workbench screen, because invisible parts can be 'dragged' into view. Panning is activated through pinching together two fingers of the Pinch Glove (the thumb and middle finger). When the user moves his arm, the scene will move as if it were attached to the user's left hand. Panning stops when the user releases the pinch. The user can only pan the scene parallel to the horizontal workbench plane. Scaling allows the user to scale the entire

scene up or down by moving the pinched fingers closer or farther from the display surface. This technique is similar to moving the viewpoint nearer or farther. We chose scaling because this ensures that the objects remain above the workbench screen and therefore within reach of the user. When objects are below the physical surface, the user should also be able to translate the scene relative to the surface.

Selection

Selection is the process of identifying one (or more) objects, usually to manipulate them in some way. It may also be used to identify a region of space and/or the objects inside. There are two general cases. First of all, the object can be within arm's reach of the user. In this case, the user can simply reach out and "touch" the object to select it (see 'Direct picking' below). Secondly, the object can be out of reach (either because it is too far away or because it is below the workbench surface). Here the user must select the object from a distance using a tool of some sort. We implemented ray casting, gaze directed selection, pointing, and virtual hands.

Direct picking. The most intuitive and easy way for the user to select an object is to simply reach out until his hand intersects the desired object. After some trial, we decided that the most precise way to support this was to put a tracker on the forefinger of each Pinch Glove (see Fig. 1) and define a small, invisible box around it. An object is selected when this box intersects the object's bounding box. The size of the box can be adjusted to allow for a larger or smaller 'error margin'. This technique is always active, even when one of the other tools described next is active as well.

Ray casting. The user can activate a ray that shoots out from the tip of his index finger. To select an object the user has to intersect it with the ray. This method is often used in VE applications to select distant objects. In our workbench environment, a ray can be activated for both hands (Fig. 2). If multiple objects are intersected, the first object is chosen.

Gaze-directed selection. When this technique is active, the user can select an object by looking at it. The user's gaze direction is approximated with an invisible ray originating between the user's eyes. Instead of pointing straight ahead, the ray points down at an angle. Trials with different angles revealed that this is more comfortable for the user because he has to look down all the time to see the workbench screen. A small cube is placed at the intersection of the ray and the workbench surface to serve as a cursor providing the user with visual feedback on his gaze direction. Again, the first object is chosen if multiple objects are intersected.

Pointing. This technique allows the user to select an object by pointing at it with his forefinger so that its tip overlaps the object in the user's field of view. To determine where the user

is pointing, an invisible ray shoots out from between the user's eyes to the fingertip. The object closest to the user's eyes is chosen.

Virtual hands. Virtual hands are virtual objects that mimic the movements of the user's real hands. When the user extends his arms, the virtual hands will extend faster than his real hands. This stretching effect allows the user to reach distant objects with the virtual hands. (This is related to the GO-GO technique of Bowman [2]). An object can be selected by positioning one of the virtual hands inside it. The function that maps virtual hand position to real hand position ensures that the user can reach any object that is visible on the workbench screen.

Manipulation

Manipulation is the act of changing certain parameters of an object (e.g. its position, orientation, shape, or color). Once an object has been selected, the user can manipulate it. Currently, only the basic manipulation tasks (translation, rotation, and uniform scaling) are supported in our application. If the object is within physical reach of the user, he can manipulate it directly with his hands. If the object is out of reach, there are three possibilities for manipulation:

- Close manipulation – bring object to user
- Distant manipulation
- Tele-manipulation – bring user to object

Close manipulation. Close manipulation brings the object to the user, after which he can manipulate it directly with his hands. In accordance with studies on the importance of proprioceptive feedback [11], we felt that close manipulation would be quite useful for detailed work on objects, especially work involving precise positioning and orientation. Close manipulation can be used whether the object is within reach or out of reach. For the former the user can pick up the object by first selecting it with the direct picking technique and then pinching together his thumb and forefinger. This will attach the object to the user's hand, allowing him to move and orient the object by moving his hand. When the object is out of reach, the user can navigate towards it by panning and scaling the scene (or walking around the workbench). We implemented two other techniques to bring an object close to the user. These are *popping* and *copying*. When active, popping causes distant objects that have been selected to 'pop' into the user's hand when he makes a pinch gesture. Copying works the same way, except that a copy of the original object is placed in the user's hand. When the user manipulates the copy, the original object will mimic the rotations of the copy. When the user releases the pinch, the copy will disappear.

Distant Manipulation. These techniques allow the user to manipulate objects using tools at a distance. The user can perform distant manipulation by selecting an object with one of

the selection tools (ray casting, gaze directed, or pointing). After this the user can make a pinch gesture, which attaches the object to the current selection tool.

Tele-manipulation. Tele-manipulation techniques let the user manipulate distant objects as if they were close to his body. The virtual hands can be used for tele-manipulation. As was described in the section on selection techniques, the virtual hands mimic the movements of the user's real hands. This allows the user to pick up a distant object with the virtual hands (Fig. 3) and then manipulate it as if he were holding the object in his real hands.

One Versus Two-Handed Manipulation

In many cases the use of two hands to perform tasks could improve performance significantly. To investigate this, we extended several of the aforementioned techniques to allow the user to use two hands. Of course the user is still able to select and manipulate different objects with either hand. Besides two-handed manipulation of objects, we implemented two-handed panning as well, since using one hand can be annoying when traversing large distances. Therefore, we changed this technique so that the user can alternate between hands (kind of like pulling himself forward along an invisible rope).

When an object is picked up with one hand, it is attached to this hand. This allows the user to move and rotate the object. However, people usually use both hands to examine an object; they pass the object back and forth between both hands and let the object 'roll' through their hands to quickly examine it from all sides. In our application, the user can pass an object from one hand to the other by selecting the object with the free hand while letting it go (releasing the pinch) with the other hand. The user can also switch to two-handed manipulation by picking an object with both hands. The object can then be rotated, translated and uniformly scaled by moving the hands relative to each other. This can be visualized by imagining a sphere between the fingertips of the user's hands. Rotation of the object can be seen as moving the fingertips across the surface of this sphere. Translation can be seen as moving the center of the sphere by moving both hands in the same direction. Finally, scaling can be seen as changing the radius by moving the hands further apart or closer together. This works the same way when the user is tele-manipulating an object using the virtual hands (Fig. 3). We changed the distant manipulation techniques (ray casting, gaze directed, and pointing) as well, but these techniques didn't work very well. This is because these techniques are like picking up an object with two sticks and manipulating it (Fig. 2).

Aids to Interaction

Constraints. When manipulating real objects, there are usually all kinds of constraints that

Workbench Characteristic	Description
Semi-immersive	Brings virtual objects into the real world; causes occlusion and boundary effects; user looks down on scene; walk around rather than walk through
Flat, limited size display	horizontal display; heights of objects above display and extents constrained; object positioning important
Physical surface	Introduces barrier to grabbing objects; requires alternatives besides direct grabbing; can provide a useful constraint
Collaboration	Naturally collaborative; hardware solutions or new interaction methods needed to handle head-tracked viewing

Table 1. Workbench characteristics affecting interaction methods and applications

restrict the user's movements (e.g. gravity, or physical surfaces like rulers, tablets or tabletops). People often depend on these things to perform precise tasks. In most virtual environments such natural restrictions are not present. However, the workbench does provide a physical screen that can be used as a physical constraint. The manipulation techniques as described above are excellent for quick examination and rough placement of objects. However, these unconstrained techniques are often not suited for more precise manipulation. Therefore, in addition to the unconstrained techniques outlined above, we implemented in-the-plane interaction, where manipulation is constrained to the region of the workbench screen.

When in-the-plane interaction is active, objects can only move parallel to the horizontal workbench surface. When the user picks up an object, it will therefore continue to rest on the surface, even when the user moves his arm up and down. When the user is holding an object with just one hand, he is only able to translate it. If he wishes to rotate or scale it, he has to grab the object with both hands. Scaling is then performed by changing the distance between the hands, and rotation by changing the positions of the hands relative to each other. Rotation is only possible about the axis perpendicular to the workbench surface.

Combining 2D and 3D

A problem in virtual environments has been the difficulty of using more traditional interface elements like windows, menus, and dialog boxes. Because these elements are two-dimensional, placing them in an environment that is 3D makes them hard to use. (The user may have difficulty finding these elements in the environment, or text on them may be hard to read from his position.) However, users have developed significant experience and skill interacting with menus and windows. The nature of the workbench (3D objects resting on a 2D surface) appears well suited for displaying such interface elements. Because the illusion of depth tends to break down at the borders of the workbench anyway, this dead space is especially suitable for displaying menu bars, icons, or information windows. We have looked into other elements from the 2D interface by implementing a simple

grid to further constrain interaction and a virtual sheet.

Grid. When the grid is activated (by pinching thumb and little finger of the left hand), horizontal and vertical gridlines are displayed on the workbench surface. This causes objects to snap to grid intersections between the gridlines. It would make sense to extend this to also cause rotation and scaling to snap to the grid. It would be possible to allow the user to adjust the grid spacing by picking up two gridlines and changing the distance between them.

Virtual Sheet. This can be seen as a sheet of paper lying on the workbench surface. The user is able to move it around by placing his hand on top of it and sliding it across the workbench. This sheet could be used to make notes, or it could be made into a floating menu.

IV. System Setup

We use the Immersive Workbench from Fakespace, Inc. It consists of an Electrohome Marquee 8500 Projection System, Polhemus 3Space Fastrak Tracking System, CrystalEyes emitter and glasses, and Fakespace Pinch Gloves. As an alternative to the Pinch Gloves, we used two button chord devices, one for each hand. The workbench is powered by a 4 processor SGI Onyx2 with IR graphics. We use three trackers: one for the user's head, and one for each hand.

Pinch Gloves. Fakespace Pinch Gloves do not recognize hand gestures, but instead they register 'pinches' (i.e. the user pressing together two or more fingers). The user can pinch two fingers of the same hand, but also one finger of each hand, or combinations of these. Obviously, this allows for a lot of different combinations. A later version of the gloves also has a big contact pad on the palms, allowing pinches that would otherwise be impossible (like forefinger and ring finger). Although intuitive, it remains to be seen if using the gloves is more intuitive than using, say, a normal button device. There are only a few pinch gestures that are really intuitive (pinching thumb and index finger to pick up an object is the best example). Other combinations are as arbitrary as assigning a mouse button to a specific command. Furthermore, a user may not be able to remember and use as many combinations as the Pinch Gloves allow.

However, it appears that people can more easily remember pinch gestures than button presses. Our qualitative observation is that this is so. Also the Pinch Gloves offer the advantage that the user can pick up and hold different tools while wearing the gloves. Disadvantages that we encountered during use were that using the keyboard becomes very difficult, and that the gloves are one-size-not-really-fits-all. Furthermore, inexperienced users often make pinch gestures unintentionally. Finally, pinches have to be precise (the contact pads on the fingertips really have to make contact). More experiments are needed testing the usability of Pinch Gloves for different tasks and comparing them with other devices.

Button chord devices. Besides the Pinch Gloves we used button chord devices, one for each hand, to provide user input. We made these from plastic tubes on which we attached a tracker and five buttons (Fig. 1). The latter are placed so that the user can access each button with a finger while comfortably holding the device. Giving commands with the button chord devices seemed to work better (a user notices fairly easily that a button is not correctly pressed down, whereas a pinch gesture can be slightly off). The button chord devices seemed to be more suitable for (distant) selection than Pinch Gloves. Because of their shape, the direction in which they are pointing is obvious to see. They also suffered less from the so-called arm lever problem. This is the problem of a small movement of the user's hand resulting in a large displacement of the tip of the selection ray. Placing the tracker on the fingertip of the Pinch Glove also provided a very clear indication of the pointing direction, but pinching causes the user's hand to move much more than pressing a button does. On the other hand the Pinch Gloves seemed to be much more suitable for close manipulation. Using the button chord devices for manipulation seemed like sticking two sticks in an object and then manipulating it, which is less natural.

V. Workbench Characteristics

One of our main goals was to find what characteristics of the workbench set it apart from other types of virtual environments. After extended evaluation, the workbench seems to have the following dominant characteristics, summarized in Table 1.

Semi-immersive. First of all, the workbench environment is not fully immersive. In HMD and CAVE environments the general goal is to completely immerse the user in the environment. In HMD environments this is accomplished by shutting off the user's field of view and presenting each eye with a small screen. CAVE systems completely surround the user with projection screens. Both these approaches result in the user being able to see only the virtual

world, although in the CAVE he can also see his body and any tools he is using. By contrast the workbench brings virtual objects into the real world rather than bringing the user into the virtual world. Instead of being inside the scene, the user is looking down on it. This difference is reflected in the interaction techniques. For example, instead of walking through the virtual environment, the user walks around the workbench to obtain different views of the scene.

Flat display of limited size. In most environments, the display screen is vertical. However, the projection surface of the workbench is horizontal or nearly horizontal. Furthermore, although the screen is large, its size is still limited. The fact that the screen is horizontal allows the user to look on top of the scene, but its limited size constrains the heights and extents of objects sticking out of the screen. If an object is not projected completely inside the screen, the user's perception of its stereo effect will tend to collapse. Therefore, objects can't be higher than the user's eye position (at least for a horizontal screen). Usually objects should be much less high than this, for example to allow the user to bend his knees to look at the side of an object without causing the object to project outside the screen. Because of this, the object(s) of interest should be positioned somewhere in front of the user, in the center of the screen. An improvement might be to have another (vertical) display behind and integrated with the workbench. This would allow the user to really look at an object from the side (and even from the bottom in some cases). It would add complication and expense to the setup, though.

Physical surface. In the workbench environment there is a physical surface directly in front of the user. In HMD setups such a surface is not present. Although CAVE systems do have physical walls and floors, they are usually relatively far away from the user. The physical surface introduces a barrier that sometimes prevents the user from grabbing virtual objects that seem to be within reach. In fact, if the interaction technique is not appropriately designed, the user will continually bang his hand into the display surface! Because of this barrier, other techniques besides direct grabbing have to be implemented, as we discussed above. In CAVE environments the goal usually is to hide the fact that the physical walls are present and to give the user the feeling that he is standing in a much larger environment than he actually is. However, for the workbench it would seem to be an advantage in many cases to use the fact that the physical barrier is there. An example of this is using the screen as a physical constraint, like we did with the constrained manipulation techniques. In other cases, for example involving detailed 3D manipulation of small objects,

Type of Interaction	Workbench Effectiveness	Comparison to Immersive
Navigation	God's eye view; mostly panning and scaling; pinch gestures are intuitive.	Navigation is often from within scene.
Selection	Objects can be inadvertently moved out of the display area & lost. Cursor necessary in case object too far above screen or too close to viewer.	Not a problem in monoscopic env.; less of a problem in CAVE. Workbench evaluations also apply to immersive envs.
Manipulation	Ray casting best for object selection; gaze-directed good for less precise selection; virtual hands effective but require learning.	Workbench evaluations also applicable to CAVE.
Close Distant	Both one & two hands especially effective. Ray casting for rough positioning; virtual hands reasonably good.	
Constrained Unconstrained	For precise placement & orientation in scene. For working on hand-held objects; two hands important here.	

Table 2. Interaction technique effectiveness

it might be better to not constrain movement along the screen. As we've discussed, the screen could be used to display 2D items like menus, grids, or information windows. [4,5] The use of these 2D interface elements presents difficulty in virtual environments, because there isn't a physical surface to restrict the user's motions. The workbench surface makes these interface elements easier to use.

Collaboration. The workbench is a naturally collaborative environment. Unfortunately the use of tracked stereo impedes two or more people working together because the untracked person will get a distorted view which can become extreme (e.g., when the tracked person moves to the opposite end of the workbench or leans well over to one side while the untracked person remains upright). Recently new methods have been developed that permit two people to be tracked simultaneously by multiplexing all four eye views. [1] However, this is expensive and does not appear scalable to more than a few people. We plan to implement and evaluate methods that may improve collaboration even when there is only one tracked person. These include virtual trading of tracking between collaborators (not physically handing off the tracker), turning tracking on and off through gesture or voice commands, and attempting to find the best averaged viewpoint for a group of users when head tracking is turned off.

VII. Interaction Technique Effectiveness

We built a simple scenario for the workbench that we used consistently in the evaluation of all interaction methods and input devices. The scenario consisted of a room with various pieces of furniture. These were positioned so that they appeared to rest on the workbench surface. We added a textured ground plane to this scene that coincided with the workbench surface, making it look opaque and solid. To prevent objects from disappearing below this ground plane, we

implemented a crude collision detection scheme. The user looks down on this scene and by using the selection and manipulation techniques he can pick up and manipulate the furniture. An object turns red to indicate selection. We implemented our test environment using the SVE toolkit, developed at the GVU Center to allow rapid development of VE applications [7]. We observed people using this scenario on the workbench during countless lab demonstrations and several planned observation sessions. In this way we observed several hundred users. This feedback permitted us to make the several updates of techniques described in this paper and is the basis for the evaluations in this section. Certainly we plan to follow up with some formal user performance studies. The results of our evaluations and observations are summarized in Table 2.

Navigation. We found that navigation does not play as big a role as in other types of virtual environments. The scene is often relatively small (model-sized), and therefore there is no need for the user to cover large distances. Head-tracking allows the user to physically walk around the workbench. This technique doesn't require the user to use a special tool for navigation, thereby keeping his hands free for other tasks. The other navigation techniques we implemented (panning and scaling) are necessary because of the limited size of the screen. Invisible portions of the scene have to be dragged into view. Using pinch gestures and arm movements to achieve this proved to be very natural and easy to use. These gestures give proprioceptive feedback as well as visual feedback so that the user quickly learns the relationship between arm position and, for example, the size of the scene.

Selection. Direct picking is a very intuitive selection technique. However, we noticed that users were having difficulty selecting objects that were floating relatively high above the workbench. This appears to be partly caused by the fact that the user receives false occlusion

cues. Because the user's physical hand is always in front of the workbench screen, virtual objects will always be projected behind it, even when the virtual object is actually placed between the user's head and hand. This causes the user to think that the object is farther away than it could in fact be closer to his eyes. In addition stereo images of objects located relatively close to the face cannot be fused, which further erodes depth information. To solve this, we put a small virtual cursor near the tip of the user's forefinger or of the button chord device. This definitely made selection easier, because the depth cues of this cursor relative to an object are always correct.

Of all distant selection techniques, ray casting seemed to work better than the others. We think that this is because with this technique the selection ray is always visible, whereas with gaze directed selection and pointing it is invisible. However, when we included the aforementioned cursor near the user's fingertip, pointing (a very intuitive technique for selecting distant objects) worked just as well. A drawback of ray casting is that, because of the limited screen size, the illusion of the ray shooting out of the user's hand breaks down when the user's hand is relatively far from the workbench (part of the ray doesn't project on the screen). A problem of the pointing technique is that the user's hand can completely obscure small objects, making selection difficult. Gaze directed selection absolutely requires the cursor. We placed this cursor at the intersection of the user's gaze direction and the workbench surface. However, this results in the user selecting an object not by simply looking at it, but by trying to place the cursor behind the object (the user doesn't focus on the object anymore, but on the cursor). Gaze directed selection is probably more suitable for selecting general regions of space (e.g., a traveling direction), instead of individual objects.

Normally, arm extension techniques like the virtual hands make selection more difficult. This is because the virtual hand has to be inside an object, whereas a ray only has to intersect the object. However, we found that our virtual hands worked pretty well, because of the limited size of the workbench. Objects are relatively close to the user, so it is not very difficult to place the virtual hand inside an object.

Manipulation. Close manipulation is without a doubt the best way to manipulate objects on the workbench. The user holds objects in his hands, close to his body, which allows him to take full advantage of proprioception. The user can rest his elbows on the edge of the workbench, roll the object through his hands, and pass it from one hand to the other. Distant manipulation (where an object is attached to, for example, the selection ray), is useful for very rough object placement. However, if a certain level of precision is required, the user should strive to bring the object right in front of him before manipulating it. If the user wishes to inspect an

object, or he has to have quick access to all sides of an object, unconstrained manipulation is the most suitable. However, if the goal is to precisely position and orient an object in the scene, the more constrained techniques should be used. When an object's movement is restricted to the workbench plane, relatively precise placement and orienting of an object is possible simply by sliding the hands across the workbench surface. Even more precise placement is possible when grids are used. Close manipulation has the drawback that small objects may get obscured by the user's hands, and the user may get false occlusion cues. Tele-manipulation (using the virtual hands) removes this problem. However, the user now seems to have four hands (because his real hands are still visible) which can be confusing. We noticed that many people had trouble using them. Because real hand movements are exaggerated by the virtual hands, they tend to quickly move around on the screen. They often even disappear from the projection screen. This was confusing for a lot of people. Also, because of this exaggeration of movement, it is difficult to keep an object at its old position while manipulating it. An improvement to the virtual hands might be to let them mimic the movements of the real hands exactly once an object has been selected for manipulation (i.e., without the scaling). However, once a user gets a little experience using them, virtual hands can be a valuable tool for manipulation, because the user can easily reach any object on the screen.

One vs. Two Hands. It appears that the proprioceptive feedback of relating one hand to the other makes fine manipulations faster and more accurate. For example if the user were attaching one object to another or placing a part in a slot in another object, it would be easier to do having an object in each hand rather than, for example, laying one object down and then using just one hand to insert the other object. Our observations seem to support this speculation. We plan user studies to confirm this idea. In addition it should be faster to, say, make multiple attachments to different sides of an object with two hands than one.

VIII. Conclusions

In this paper we have studied and described the dominant characteristics of the workbench. These characteristics have to do with its semi-immersive nature, its flat, limited size display that presents a physical barrier, and its natural capability for collaboration. Interaction method effectiveness is affected by these characteristics. We discuss effectiveness in the three main interaction categories of navigation, selection, and manipulation, concentrating mainly on the latter two. Based on observations and demos involving several hundred users, we have evaluated and adjusted many interaction techniques. We find that although there are

problems caused by limited screen space size and stereoscopic display, these can be minimized with proper cues and adjustments. The results on interaction methods described here should be appropriate for a wide variety of applications.

We plan to follow up with controlled user studies for a number of interaction techniques where we will be able to quantitatively evaluate performance. In particular we will look at two-handed manipulation applied to detailed attaching and inserting of things on 3D objects. We will also look at the possibility of finding new interaction and display methods to minimize head-tracked viewing problems among multiple collaborators on the workbench.

Acknowledgments

This work was performed in part under contracts from the Naval Research Lab and the Office of Naval Research. We thank Frank Jiang for help in setting up the workbench environment and creating images for this paper.

References

1. M. Agrawal, A.C. Beers, B. Frohlich, P. Hanrahan, I. McDowall, and M. Bolas (1997). The Two-User Responsive Workbench: Support for Collaboration Through the Individual Views of a Shared Space. Computer Graphics Proceedings, ACM SIGGRAPH 1997, pp. 327-332.
2. D. Bowman, D. and L. Hodges (1997). An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. Proceedings of IEEE VRAIS '97, pp. 35-38.
3. C. Cruz-Neira, D.J. Sandin, and T.A. Defanti (1993). Surround-Screen Projection-based Virtual Reality: the Design and Implementation of the CAVE. Computer Graphics Proceedings, ACM SIGGRAPH 1993, pp. 135-142.
4. L.D. Cutler, B. Frohlich, P. Hanrahan (1997) Two-Handed Direct Manipulation on the Responsive Workbench. 1997 Symposium on Interactive 3D Graphics.
5. J. Durbin, J.E. Swan, B. Colbert, J. Crowe, R. King, T. King, C. Scannell, Z. Wartell, and T. Welsh (1998). Battlefield Visualization on the Responsive Workbench. To be published, Proceedings of IEEE Visualization '98.
6. Yves Guiard (1987). Symmetric Division of Labor in Human Skilled Bimanual Action: the Kinematic Chain as a Model. Journal of Motor Behavior 19(4), pp 486-517).
7. Drew Kessler, Rob Kooper, and Larry Hodges (1998). Simple Virtual Environment Library: Version 2.0. Georgia Tech, GVU Tech Report 98-13.
8. W. Krueger, C. Bohn, B. Frohlich, H. Schuth, W. Strauss, and G. Wesche (1995). The Responsive Workbench: A Virtual Work Environment. IEEE Computer, pp. 42-48.
9. J. Leigh, A.E. Johnson, C.A. Vasilakis, and T.A. DeFanti (1996). Multi-Perspective Collaborative Design in Persistent Networked Virtual Environments. Proceedings of IEEE VRAIS '96, pp. 253-260.
10. M.R. Mine (1995). Virtual Environment Interaction Techniques. University of North Carolina, Tech Report TR95-018.
11. M.R. Mine, F.P. Brooks Jr., C.H. Sequin (1997). Moving Objects In Space: Exploiting Proprioception In Virtual-Environment Interaction. Computer Graphics Proceedings, ACM SIGGRAPH 1997, pp. 19-26.
12. U. Obeysekare, C. Williams, J. Durbin, L. Rosenberg, R. Rosenberg, F. Grinstein, R. Ramamurthi, A. Landsberg, and W. Sandberg (1996). Virtual Workbench - A Non-Immersive Virtual Environment for Visualizing and Interacting with 3D Objects for Scientific Visualization. Proceedings of IEEE Visualization '96., pp. 345-349.



Fig. 1 Pinch Glove and Button Chord input devices showing placement of trackers.

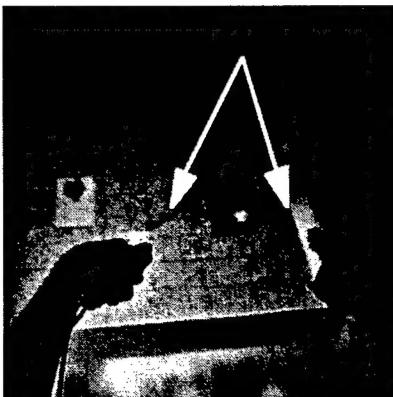


Fig. 2 Manipulation of object using ray casting (indicated by arrows).

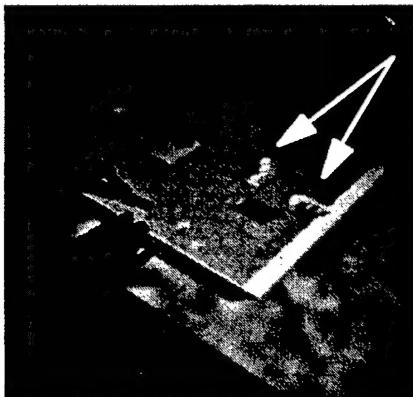


Fig. 3 Manipulation of object using virtual hands (indicated by arrows). The evaluation scenario can be seen.